CLAIMS

What is Claimed, is:

1. A method of transferring data from a hardware device comprising:

a) allocating non-pageable memory to a requesting program; and

b) transferring via a direct memory access (DMA) from a hardware device into said non-pageable memory.

2. The method of Claim 1, further comprising:

an application program requesting said allocation of said non-pageable memory that is accessible to said application program.

3. The method of Claim 1, further comprising:

c) an application program reading said data from said non-pageable memory.

4. The method of Claim 3, wherein said reading said data from said non-pageable memory comprises transferring said data directly from said non-pageable memory to a buffer that is accessible to said application program.

5. The method of Claim 1, wherein said a) comprises a driver receiving a request for said non-pageable memory that is accessible by said requesting program.

6. The method of Claim 1, further comprising a driver notifying said requesting program when said DMA is complete.

7. The method of Claim 1, wherein said requesting said allocation of non-pageable memory comprises specifying a range of address to be used for DMA transfers from said hardware device.

8. The method of Claim 1, wherein said hardware device is a graphics device.

9. The method of Claim 1, wherein said hardware device is a network device.

10. The method of Claim 1, wherein said data is pixel data.

11. The method of Claim 1, wherein said data is texel data.

12. The method of Claim 1, wherein said data is vertex data.

13. The method of Claim 1, wherein said DMA is performed using an address re-mapping table.

14. The method of Claim 1, wherein said non-pageable memory is within a graphics aperture re-mapping table (GART) address space.

15. The method of Claim 1, wherein said non-pageable memory is AGP (Accelerated Graphics Port) memory.

16. The method of Claim 1, wherein said non-pageable memory is PCI (Peripheral Component Interconnect) memory.

17. The method of Claim 1, wherein said DMA comprises central processing unit (CPU) cache snooping.

18. The method of Claim 1, wherein said b) comprises a driver requesting a direct memory access (DMA) data transfer from said hardware device into said non-pageable memory in response to receiving a request for said data transfer.

19. A method of controlling data transfer from a hardware device, comprising:

receiving a request from an application program for a data transfer, said request specifying a range of addresses in which to transfer said data;

determining whether said range of addresses is within a pre-allocated range of non-pageable memory; and

if said range of addresses specified in said request is within said pre-allocated range of non-pageable memory, requesting a DMA between a hardware device and said range of addresses, wherein said data is transferable directly between said application program and said range of addresses in said pre-allocated non-pageable memory.

20. The method of Claim 19, wherein if said range of addresses specified in said request is within said pre-allocated range of non-pageable memory, said

data is available to said application program to read directly from within said non-pageable memory.

21. The method of Claim 19, wherein if said range of addresses specified in said request is within said pre-allocated range of non-pageable memory, said data is transferable directly from said application program to said range of addresses in said pre-allocated non-pageable memory.

22. The method of Claim 19, further comprising:

if said range of addresses specified in said request is not within said pre-allocated range of non-pageable memory:

requesting a DMA of said data from said hardware device to system memory;

transferring said data from said system memory to a buffer; and

transferring said data from said buffer to said range of addresses specified in said request, wherein said data is available to said application program.

23. The method of Claim 19, further comprising:

if said range of addresses specified in said request is not within said pre-allocated range of non-pageable memory:

transferring said data from said range of addresses specified in said request to a buffer;

transferring said data from said buffer to system memory; and

requesting a DMA of said data from said system memory to said hardware device.

24. The method of Claim 19, wherein said determining whether said range of addresses is within said pre-allocated range of non-pageable memory comprises determining whether said range of addresses is within a range of address specified by said application program as for read DMA operations.

25. The method of Claim 19, wherein said determining whether said range of addresses is within said pre-allocated range of non-pageable memory comprises determining whether said range of addresses is within a range of address specified by said application program as for write DMA operations.

26. The method of Claim 19, wherein said hardware device is a graphics device.

27. The method of Claim 19, wherein said hardware device is a network device.

28. The method of Claim 19, wherein said data is pixel data.

29. The method of Claim 19, wherein said data is texel data.

30. The method of Claim 19, wherein said data is vertex data.

31. The method of Claim 19, further comprising:

if said range of addresses in said request is within said pre-allocated range of non-pageable memory, transferring said data directly between said range of addresses within said non-pageable memory and a buffer that is accessible to said application program.

32. The method of Claim 19, wherein said DMA comprises using an address re-mapping table.

33. The method of Claim 19, wherein said non-pageable memory is within a graphics aperture re-mapping table (GART) address space.

34. The method of Claim 19, wherein said non-pageable memory is AGP (Accelerated Graphics Port) memory.

35. The method of Claim 19, wherein said non-pageable memory is PCI (Peripheral Component Interconnect) memory.

36. The method of Claim 19, wherein said DMA includes central processing unit (CPU) cache snooping.

37. The method of Claim 19, wherein said DMA is performed using an address re-mapping table.

38. A system for transferring data, comprising:

a first processing unit;

a memory coupled to said first processing unit; and

a second processing unit coupled to said memory;

wherein said system is operable to:

allocate, in response to a request from an application program, non-pageable memory in said memory, wherein said non-pageable memory is accessible by said application program;

initiate a transfer of data via a direct memory access (DMA) from said second processing unit into said non-pageable memory in response to a request from said application program; and

transfer said data from said non-pageable memory to a buffer in response to a request from said application program.

39. The system of Claim 38, wherein second processing unit comprises a graphical processor unit (GPU).

40. The system of Claim 38, wherein second processing unit comprises a network processor unit (NPU).

41. The system of Claim 38, wherein said system is further operable to transfer said data directly from said non-pageable memory to said buffer.

42. The system of Claim 38, wherein said DMA uses an address re-mapping table.

43. A system for transferring data, comprising:

a first processing unit;

a memory coupled to said first processing unit; and

a second processing unit coupled to said memory;

wherein said system is operable to:

receive a request from an application program for a data transfer involving said second processing unit;

determine whether a range of addresses specified in said request is within a pre-allocated range of non-pageable memory; and

if said range of addresses specified in said request is within said pre-allocated range of non-pageable memory, request a direct memory access (DMA) of said data between said second processing unit and said range of addresses, wherein said data is transferable directly between said application program and said range of addresses in said pre-allocated non-pageable memory.